Where can I do this? Geometric Affordances from a Single Example with the Interaction Tensor

Eduardo Ruiz and Walterio Mayol-Cuevas

***Abstract*— This paper introduces and evaluates a new tensor field representation to express the *geometric* affordance of one object relative to another, a key competence for Cognitive and Autonomous robots. We expand the bisector surface representation to one that is weight-driven and that retains the provenance of surface points with directional vectors. We also incorporate the notion of *affordance keypoints* which allow for faster decisions at a point of query and with a compact and straightforward descriptor. Using a single interaction example, we are able to generalize to previously-unseen scenarios; both synthetic and also real scenes captured with RGB-D sensors. Evaluations also include crowdsourcing comparisons that confirm the validity of our affordance proposals, which agree on average 84% of the time with human judgments, that is 20-40% better than the baseline methods.**

I. INTRODUCTION

A key objective of Robotics is to devise systems that can operate in previously unknown environments. This has so far been a stumbling block for autonomous and cognitive robots, which largely fall back onto either well scripted scenarios, or scenarios for which the amount of training needed, undermines the notion of little prior knowledge. A different approach has been to aim for one of the most elusive concepts to date in the perception-action coupling, that is, the concept of affordances.

The notion of affordances posed by J.J. Gibson [1], calls for an approach to visual perception that is there to help the perceiving agent to interact with the world. Specifically, visual perception is described as a process to understand what can be done where. Which is fundamentally different to asking the two separate questions of "what is this?" and subsequently asking "how can I use it?". Such an unified representation of the world is immediately useful as by definition it is one that already takes into account what the agent is capable of.

Furthermore, Gibson also argued that affordances are "immediate" to perceive. This has often been misread as a call to ignore the relevance of the representation [2]. But we argue that such direct affordance perception rather motivates methods that are able to immediately transfer what has been learned to other objects and places after a small number or even a single observation of the affordance.

Being able to determine affordances can have profound implications for acting-perceiving agents. It can in principle liberate the computational approach to visual processing from the focus on objects and their arbitrary labels which have to be extensively learned. To learn an affordance is not to classify an object [1], since a cup is not only for drinking but also a paperweight, or even a tool to build sand castles. Our key motivational insight here is that an affordance is necessarily the result of the composition between the world and the agent. Understanding and modeling this interaction between objects and the world is the central focus of our work.

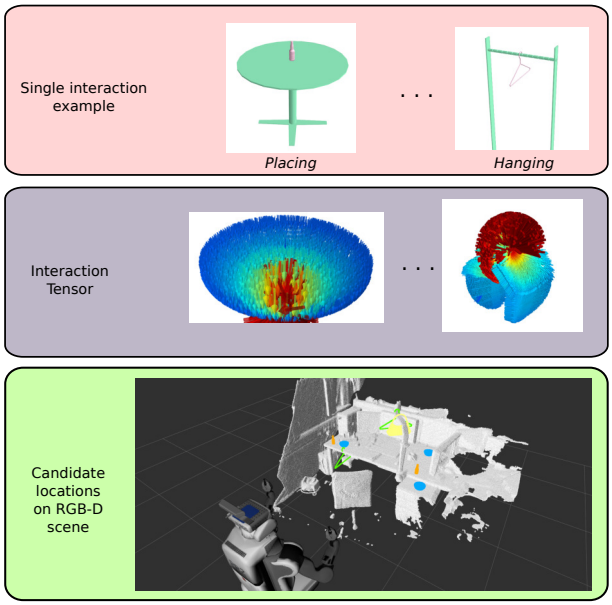


Fig. 1: Our affordance tensor allows the prediction of candidate locations for interactions on real RGB-D data. Using our approach, a robot can go into an unknown scene and answer perceptual questions like: *where can I hang a handbag? (yellow) or a coat-hanger? (green), where can I place a bottle? (orange) or a bowl? (blue)*. All this from a single interaction example per affordance obtained with different and synthetic 3D models.

We concentrate on the subclass of affordances between rigid objects. Affordances such as "where can I hang this?", place this, ride, fill, and similar. We do this by specifying a geometry-driven interaction tensor that aims to capture the *way* in which the affordance manifests between a pair of objects. We therefore do not consider dynamic affordances nor planning subtasks to achieve them. Fig. 1 and Fig. 2 depict a general view of our approach and examples of the interaction tensor respectively.

Importantly, as a departure from prior methods, using only

The authors are with the Department of Computer Science, University of Bristol, UK er13827@bristol.ac.uk, wmayol@cs.bris.ac.uk

E.R. thanks the Mexican Council of Science and Technology (CONACYT) for sponsoring his studies with the scholarship number 399136.

a single example we detect other viable places for such geometric affordances in previously unseen locations. Our evaluation corroborates our approach with both synthetic and real scenes.

Our contributions in this paper can be outlined as follows:

- We extend the notion of the bisector surface to a weighted vector field —an interaction tensor field.
- Show how this tensor with direct, sparse sampling, allows for the determination of geometrically similar interactions even from a single example, and is better than existing formulations.
- Introduce the notion of *affordance keypoints* which serve to more quickly judge the likelihood of an affordance at a test point.
- Evaluate with both synthetic and real scenes from RGB-D mapped areas.
- We validate results with crowdsourced judgments.

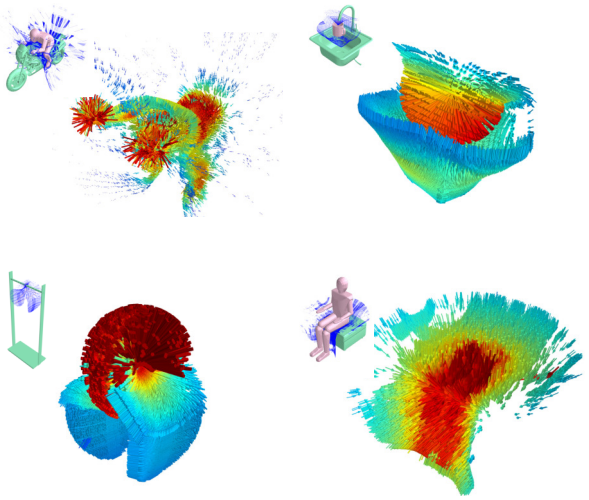


Fig. 2: Interaction tensor examples of 4 affordances. Starting from the top-left in clockwise direction: *riding* a motorcycle, *filling* a mug, *sitting* and *hanging* a coat hanger.

II. RELATED WORK

Affordance detection has been studied in recent years in both Computer Vision and Robotics. Briefly speaking, affordance knowledge has been incorporated in learning systems that use data from demonstrations of interaction, robot self exploration and static labeled imagery. In terms of the applications, the approaches include semantic scene understanding, grasp learning, gesture recognition, object segmentation and planning in goal-directed tasks. An important body of research comes from the developmental robotics field [3]. The core of these approaches is the representation and learning of actions and their consequences over a set of objects, which can be applied to action selection and planning[4] or tool selection for achieving a certain task [5]. These approaches use visual features describing shape, color, size and relative distances to capture object properties and effects. Using robot self-exploration and human demonstrations the systems benefit from single-object affordances to execute more complex interactions and execute a plan (task planning). For instance, [6] shows a robot learning in a self-supervised manner to use a tool by observing the effects of its actions on other objects.

Another line of research that has benefited from affordance learning is Human-Robot Interaction [7], [8], [9], [10], [11], [12], [13]. In these studies the main goal is to to perform action recognition in a robot observing humans, usually to predict or anticipate human activities, and in this way assist humans better while they perform everyday tasks.

Work has also been done using static imagery, where the affordance or interaction is provided as a label rather than demonstrated. [14], [11], [15], [16] based their work on labeled 2D imagery to predict functional regions or attributes on every day objects. Other works exploit 3D information to learn and predict affordances of objects in the environment. In [17], the concept of 0-order affordance is introduced to refer *hidden* affordances that can be found on an object but not in its current pose. Amongst the affordances studied are rollable, containment, liquid-containment, unstable, stackable-onto and sittable. In [18] a physics-based simulation on CAD models of objects is used to learn three functional classes: drinking vessel, table and sittable. Using geometric features on RGB-D data [19] presents a segmentation algorithm that learns and predicts affordances such as pushable, liftable and graspable on indoor scenes. In [20], [21] RGB-D images are used to learn and predict functional regions such as grasp, contain, support and cut on objects placed on a table-top. Using RGB-D images of indoor scenes [22] perform segmentation for human actions such as walkable, sittable, lyable. Similarly, affordances are studied in [23], [24], [13] to map locations suitable for sitting, or laying down; particularly in these cases using human skeleton *hallucinated* on the different indoor scenes.

Crucially, these previous methods are heavy in terms of requiring multiple learning examples, impose a particular parameterization such as detection of planes or shapes and or are highly specific to an object e.g. humanoid shapes. Our approach aims to address various of these limitations, namely relying on pre-parameterization of scene or objects and relying in numerous examples.

An interesting additional related work is [25], where an algorithm for 3D scene indexing is developed to capture hierarchical relationships among objects using Betti numbers [25]. It proposes Interaction Bisector Surface curvature descriptors that are learned from multiple examples. There, it is shown the discriminative power of the Interaction Bisector Surface (IBS) to characterize the relationships between sets of objects. The bisector surface B of two objects O_1, O_2 is the locus of points equidistant to the objects' surface. The bisector surface is an approximation of the Voronoi diagram for objects in a scene. We extend the robustness of the IBS by preserving information regarding the expected locations or areas in the 3D space that enable the interaction. This is what we call the Interaction Tensor (**iT**).

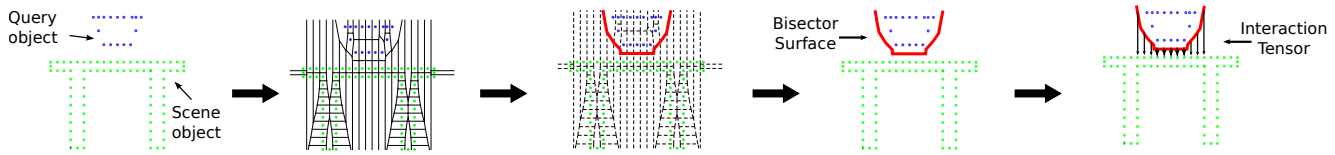


Fig. 3: The interaction tensor is computed from the bisector surface. First, objects are placed simulating the interaction. The Voronoi diagram is calculated amongst all the data points. Only ridges splitting points from different objects are taken into account. These points comprise the bisector surface (red), which is used to compute the interaction tensor for *placing* a bowl on a table.

III. OUR APPROACH

Briefly speaking, our method consists on computing the **iT** descriptor between a pair of objects of whose affordance is being investigated. Examples of the **iT** between pairs of objects are shown in Fig. 2. The method allows us to use a model of say a humanoid *skeleton* and predict *human affordances* such as sitting; similarly to [24], [13], [23], [22]. But importantly, it also allows us to build these tensors more generally for any other pair of objects such as a coat hanger that needs to be placed on the (unknown) scene.

In our approach, we refer to the two interacting objects as **query-object** and **scene-object** (or scene) respectively. The query-object is the one with a known affordance; a mug which affords *filling*, is an example of a query-object in our setup. A scene-object is the second part of the interaction; this could be a second object or part of a scene or furniture that allows the affordance to take place. Using the same mug filling example, a faucet or tap and sink would act as scene-object.

A. Computing the Interaction Tensor

We start by computing the IBS between a pair of objects similarly to [25]. Using 3D or CAD models of the interacting objects, the first step is to create dense point clouds by uniformly sampling points on the surfaces of the models. The objects are placed relative to each other simulating the interaction that they would have on real circumstances (affordance *training* example). Once these objects are in the desired positions, the Voronoi diagram is computed for the complete pointcloud comprising both objects; this produces a simplicial complex where polygon ridges are equidistant to the points that produced them. The IBS is comprised of ridges shared by points from different objects. Additionally, we preserve the vector(s) that contributed to the computation of a given point in the IBS, that is what we called **provenance vectors**. This process generates the Interaction Tensor for the affordance simulated by the two interacting objects. Note that the provenance vectors should not be confused with surface normal vectors on the IBS, since the latter do not provide information regarding the origin of points in the IBS.

In principle, the IBS and **iT** extend towards infinity; in practice, we trim these to fit a sphere of radius equal to the diagonal of the query-object bounding box. Fig. 3 illustrates how the interaction tensor is computed from the bisector surface between two sets of points. Specifically, it shows the **iT** for *placing* a bowl on a table in a simplified 2D scenario.

Formally, given the set of points on the bisector surface B and the scene-object represented by the set of points O, the tensor field characterizing the interaction is defined as

$$\mathbf{iT}(B,O) = P\hat{i} + Q\hat{j} + R\hat{k} \tag{1}$$

where $\hat{i}$,$\hat{j}$,$\hat{k}$ are the unit vectors in the direction of the x, y, and z axes of a three dimensional Cartesian coordinate system and where

$$P = \hat{G}(B,O)_{\hat{i}} - B_{\hat{i}}$$

$$Q = \hat{G}(B,O)_{\hat{j}} - B_{\hat{j}}$$

$$R = \hat{G}(B,O)_{\hat{k}} - B_{\hat{k}}$$

with

$$\hat{G}(B,O) = \underset{o_i \in O}{\operatorname{argmin}} \|o - B\|_2$$

The interaction tensor inherits from the bisector surface the discriminative power in characterizing the relationships between sets of objects. It preserves key geometrical features while being robust to changes in the geometry of the interacting objects. Figure 4 shows interaction tensor examples generated using the same query-object (coat hanger) and scene-objects (coat racks) with varying geometries. In Fig.8, the same single example affordance tensor learned from the synthetic scene is used on a real RGB-D scene where meaningful placements are proposed. These figures demonstrate that despite geometrical changes in the interacting objects the **iT** retains the overall shape or geometrical features characterizing the interaction.

A single **iT** example is computed for every affordance considered in our research: *placing*, *hanging*, *filling*, *sitting* and *riding*. Fig. 1 and Fig. 2 show the interaction examples and tensors used for our experiments.

B. Weighted Interaction Tensor

Every point in the bisector surface is defined by a set *provenance vectors* $P = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_i\}$, we use this information to assign a weight $W = \{w_1, w_2, \ldots, w_i\}$ to every location on the interaction tensor

Assuming that the scene-object pointcloud is dense enough, we can simply take one of such vectors without losing generality. The weight related to a point in the interaction tensor is computed from the magnitude of its

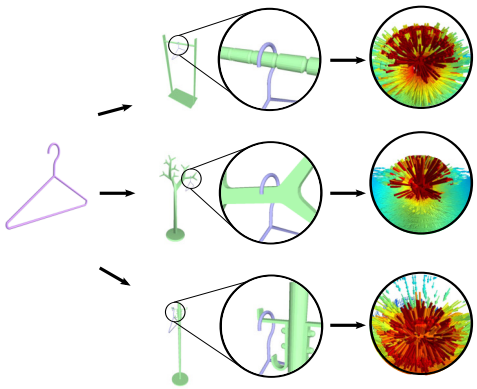


Fig. 4: Interaction tensor for *hanging* a coat hanger on racks with different geometries. Although changes occur in specific locations of the tensor, the key features of the interaction are preserved.

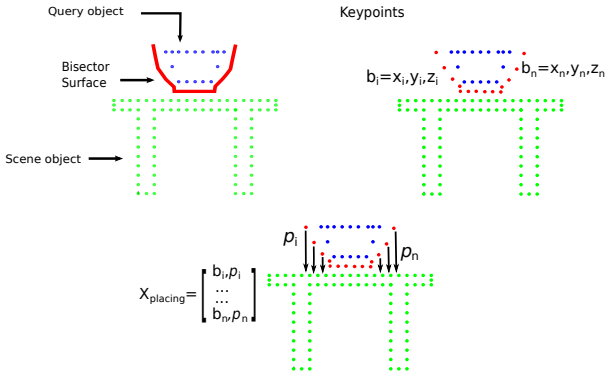


Fig. 5: Affordance descriptor for *placing* a bowl on a table in a 2D scenario. A set of points is sampled from the bisector surface. An affordance keypoint is obtained by computing the interaction tensor over these sampled points. These keypoints lead to the interaction tensor descriptor X_{placing}.

corresponding *provenance vector*. This weight or distance, represents how relevant every point is for the interaction taking place between the objects.

$$w_i = 1 - \frac{|\vec{p_i}| - |\vec{p}_{\min}|}{|\vec{p}_{\max}| - |\vec{p}_{\min}|} \tag{2}$$

Equation (2) assigns higher weights to *provenance vectors* with smaller magnitudes and vice versa to larger *provenance vectors*, while at the same time maps these weights into the range [0,1]. The idea behind this weighting method is to assign high weight to locations in the **iT** that are highly relevant for the interaction. These typically are locations where objects come closer together or touch, for instance the hook and rail area in the Fig. 4.

Fig. 1, 2 and 4 depict the weights as the color of every vector in the interaction tensor. High weights are colored in red while lower weight locations are rendered in blue.

The **iT** is a high dimensional and rich representation for object interactions, employing it directly as descriptor for affordance prediction would require costly computational resources. In order to reduce computational costs and improve the generalization capabilities of the descriptor, we reduce dimensionality by drawing N samples from **iT** (N=512 in our experiments). This subset comprises what we call *affordance keypoints* $X = \{X_1, X_2, ..., X_n\}$ where $X_i = \langle b_i, \mathbf{iT}(b_i, O)\rangle$. This lower-dimensional descriptor is formed by a set of points on the bisector surface and their corresponding **provenance vectors**. In other words, each *affordance keypoint* is formed by a 6-dimensional feature vector which consists of the x, y, z coordinates of the data point b_i on the bisector surface, and the *provenance vector* $\vec{p}$ to its nearest neighbor in the scene-object $\mathbf{iT}(b_i, O)$. Fig. 5 depicts graphically the method to compute *affordance keypoints* forming the descriptor for *placing* a bowl on a table in a 2D case.

C. Affordance query

We are interested in predicting affordances or interaction possibilities on an input scene. Given a query-object and an affordance of interest, we predict good locations or candidate places in the scene where the interaction could take place. Examples of such testing scenario are: "where can I place a bottle?","where can I hang a handbag? or "where can I fill a mug?"

Using these type of questions we perform a search over the input scene. Whereas this could be seen as an exhaustive process, but there are ways in which we can speed things up as discussed later.

In order to make affordance location predictions we follow Algorithm 1. First, test points $T = \{t_1, t_2, \ldots, t_n\}$ are uniformly sampled all over the input scene (30% of the total scene pointcloud in our experiments). Then, we extract a voxel centered at a test point t_i with a radius r_o equal to the diagonal of the bounding box surrounding the query-object. From the *training* example we have an approximation of the pose of $X_{\text{affordance}}$ relative to the scene-object. We apply such transformation to $X_{\text{affordance}}$ at test time in order to *align* it as it would be expected if the interaction could take place at t_i. Using the pointcloud of the current voxel as scene-object, a nearest-neighbor search is performed for every keypoint in $X_{\text{affordance}}$. With the nearest-neighbor search, vectors $\vec{v_t}$ are computed at test time (i.e. online), these are an approximation of *provenance vectors* found in the **iT** *training* example. In other words, each one of these test vectors goes from a keypoint in the descriptor to its nearest-neighbor in the voxel. Test vectors and example *provenance vectors* are compared to obtain a score s_i for the current pose. We compute this score at different orientations θ (8 orientations evenly distributed in $[0, 2\pi)$ in our experiments), which are obtained by spinning $X_{\text{affordance}}$ around the gravity vector centered at the current test point.

First we report on using empirically tuned threshold $S_{\text{prediction}}$ used to detect good affordance predictions with the most likely orientation of the query-object. In subsection IV-B, we introduce the value for the detection threshold that leads to the optimal performance.

The function to compute the *alignment* quality (i.e. score) at a particular location, given test vectors $\vec{v_t}$ and *training*

Algorithm 1 Affordance query

for all test points T in scene **do**
  Extract voxel of radius d_o around t_i
  for all orientations θ **do**
    Estimate test vectors using NN-search
    Compute score s_i at θ_i using (3)
    if $s_i \geq S_{\text{prediction}}$ **then**
      Predict good location at (t_i,θ_i) with probability s_i

provenance-vectors $\vec{p}$, is as follows

$$s_i = \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi w_i^2}} e^{-\frac{\Delta_i^2}{2w_i^2}} \tag{3}$$

where

$$\Delta_i = \frac{\|\vec{v}_{ti} - \vec{p}_i\|}{\|\vec{p}_i\|}$$

Where Δ is the magnitude of the difference between vectors as a proportion of the expected provenance vector $\vec{p_i}$. Briefly speaking, each comparison between vectors amounts towards the probability estimate (i.e. score) of a test point. As can be seen in (3), this is equivalent to fitting a Gaussian distribution to the difference between vectors, where the *acceptable* variance changes according to the keypoint's weight w_i.

As noted earlier, we perform a search on test points located all over the scene due to the fact that we want to remain agnostic about complex features on objects or surfaces in the scene that enable the affordance. In order to perform this search efficiently we implemented the most computationally expensive parts of our code on a GPU. With this implementation we can perform nearest-neighbor search, vectors comparison and scores computation for all orientations in 10 ms on average using a NVIDIA Titan X GPU.

IV. EXPERIMENTAL RESULTS

A. Synthetic data

For our experiments in synthetic data, we considered a total of fifteen synthetic scenes: 5 living rooms, 5 kitchens and 5 offices; and 8 affordance-object pairs *filling-mug, filling-cup, placing-bottle, placing-bowl, hanging-hanger, hanging-handbag, sitting-human, riding-human*. All the CAD models (objects and scenes) were publicly available from the Trimble 3D warehouse [1].

Examples of these scenes are shown in the figures from the following subsections, due to space limitations we only show results from a subset of our scene dataset. However, more data is available upon request. We also suggest watching the accompanying video.

[1]https://3dwarehouse.sketchup.com/

B. Evaluation

1) Interaction Tensor vs baselines: First we compare the performance of our approach against using the IBS as descriptor. For this baseline comparison a score is computed between the IBS from the interaction example and the one computed at test time using ICP ([26] implementation). In addition to being slower or more computationally intensive, the IBS descriptor is much more strict by trying to find only interaction opportunities closely similar to the *training* example. One first advantage of our approach is that, by considering a weighted vector field, we have a more relaxed matching criterion in parts of the interaction that are not critical to the affordance; this allows us to detect affordance locations in spite of variations in the scene geometry while remaining robust against false positives. In order to achieve a performance similar to the **iT** descriptor, it is necessary to relax the matching threshold for IBS comparisons; however, this increases the number of false positives. Fig. 6c shows an example of such circumstances for *hanging* a coat-hanger on a rack.

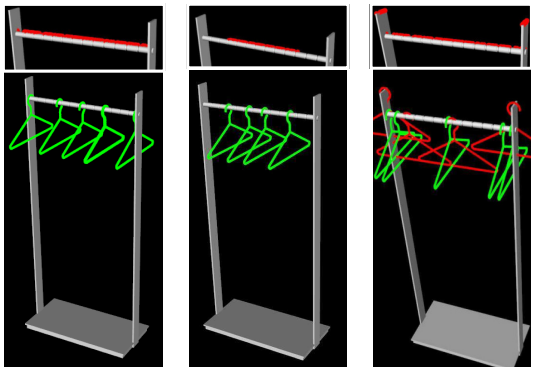
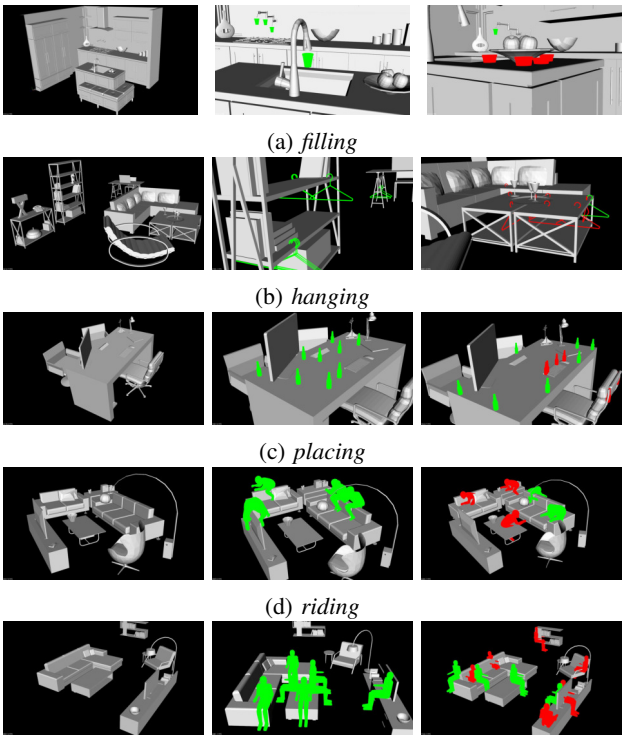


Fig. 6: The **iT** descriptor (a) allows more flexibility in the prediction of affordance location candidates. The IBS (b) predicts affordance location closely similar to the *training* example (center of the hanging rack). In order to achieve similar performance with IBS the similarity threshold has to be relaxed (c), but at the expense of increasing the number of false positives (red coat hangers).

We then evaluated and compared our results against another baseline algorithm that we call *Naive*. This algorithm simply computes pairwise distances between the query-object and the scene-object, but without any explicit representation of the interaction between the objects; therefore the goal is to find the best possible alignment at test time using the score of the alignment in the interaction example as matching criteria. This is somewhat representative of methods that use object instances as examples instead of instances representing the interaction between objects. Fig. 7 shows results contrasting the *Naive* algorithm and our approach. For fairness, both of the baseline algorithms sample points uniformly from all over the scene similarly as we do in our approach.

One thing to notice about the *Naive* approach is that it



(a) *filling*

(b) *hanging*

(c) *placing*

(d) *riding*

(e) *sitting*

Fig. 7: Affordance predictions. Results on the center column show predicted positions using the **iT** descriptor. Results in the column on the right show predictions made with the baseline *Naive* algorithm. *Naive* algorithm predicts good locations with equal probability as bad or unachievable configurations (red).

does find some expected locations; however it also predicts as good, the locations with object penetrations, occlusions or intersections; these kind of predictions would not be useful or achievable in reality. For instance 7e and 7d show *Naive* predictions for *sitting* and *riding* where the legs or parts of the body (query-object) are inside furniture. Similar cases are observed in Fig. 7a - 7c, where the predicted locations would make the query-object collide or to be inside other objects in the scene.

C. Humans criteria

To further evaluate the affordance prediction results, Amazon Mechanical Turk was employed to investigate the performance of our method according to human criteria. There is an intrinsically subjective aspect in affordance determination. We do not assume that specific objects in the scene afford interactions, we predict affordance locations candidates which eventually the agent would choose to accomplish in an action. As an example, one can afford to place a bowl on a chair as much as one can sit on the kitchen's table. These are arguably valid placings but we need humans to validate these instances.

Human "annotators" were asked to select good locations for each one of the 5 affordances considered in our research. People were presented with 6 different location candidates at a time; they had to choose amongst these options, the ones that according to them were good locations for the interaction to take place. A total of 60 persons were involved in this "annotation" of affordance locations, each person provided 10 annotations per affordance. Using the consensus of human annotations as ground truth we compute performance metrics for our approach and the baselines. Results of this evaluation are shown in Table I, which shows that on average our approach achieves an accuracy of 84.90% and f-score of 82.59%; outperforming the baseline methods in nearly all the affordance predictions. In other words, using a single example, our method consistently predicts top geometric affordance locations in unseen areas that agree with human criteria approximately 85 percent of the time.

	iT		Naive		IBS	
	Accuracy	F-score	Accuracy	F-score	Accuracy	F-score
placing	80.30	**81.16**	66.67	66.67	66.67	40.00
sitting	72.00	63.16	77.78	50.00	88.89	**66.67**
filling	96.00	**96.30**	87.50	88.89	87.50	90.91
riding	90.48	**90.00**	75.00	75.00	37.50	54.55
hanging	85.71	82.35	75.00	80.00	37.50	54.55
Average	**84.90**	**82.59**	76.39	71.48	63.61	61.34

TABLE I: Affordance prediction performance evaluated according to human annotators criteria (in terms percentage).

It is worth noticing that in the case of *riding*, which could be considered the most complex interaction, **iT** outperforms the baseline methods with a more significant difference over IBS. As discussed previously, the baseline IBS algorithm mainly detects affordance at locations with scene geometries very close to the example; since there is no motorbike-like geometry it struggles to predict such affordance. A similar situations occurs for *placing* affordances, which remains challenging for the baseline algorithms. We believe this is due to the scene geometry; for instance, baseline algorithms will not *place* the query-object if the area is not completely clear (flat clear surface). All the algorithms have a high performance with *filling*. We believe this is mainly due to the distinctive geometry of faucets and sinks, which are usually found very seldom (one in most kitchen scenes) and this makes easier to correctly detect the *filling* affordance. Another remarkable result is *hanging*; according to: human criteria, **iT** and Naive, hanging a coat hanger on edges of flat surfaces is regarded as possible. Traditional methods based on object appearance would fail to detect these cases.

As seen in the previous table, our approach outperforms the baselines most of the times in the individual affordance prediction task. We are interested in getting a a single value for the detection threshold that yields the best performance in the general affordance prediction. We used the human labeled data to obtained the value of this parameter. The results from this process is shown in Fig. 9, where the best performance for all 5 affordances is obtained with a threshold value of 0.52. In other words, a prediction of our algorithm with a score above 0.52 will be accurate 82% of the time, regardless of the affordance being queried. It interesting to see that in some points the baselines perform worse than random. When considering the every-affordance case, the overall performance of the baselines is penalized or heavily

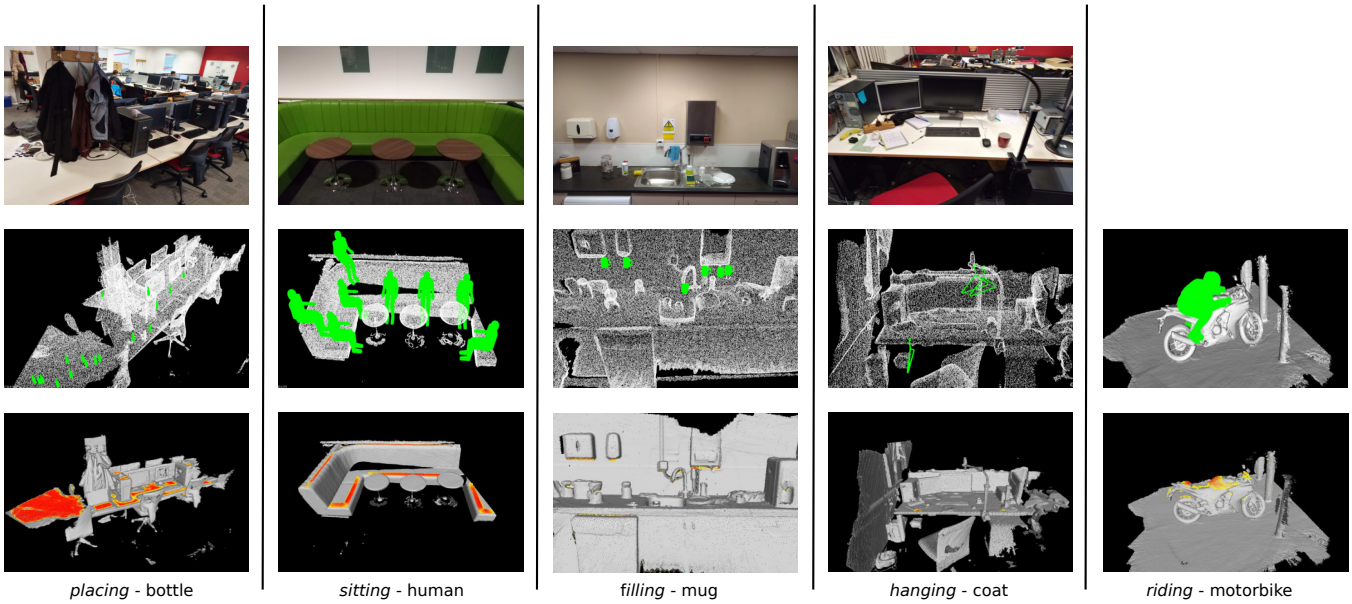


Fig. 8: Affordance heatmap with predicted locations in RGB-D scenes. From left to right: *placing* a bottle in office environment, *sitting* in reading room, *filling* a mug in kitchen, *hanging* coat hanger in office desk and *riding* motorcycle.

influenced by the low performance on complex affordances such as *hanging* or *riding*.

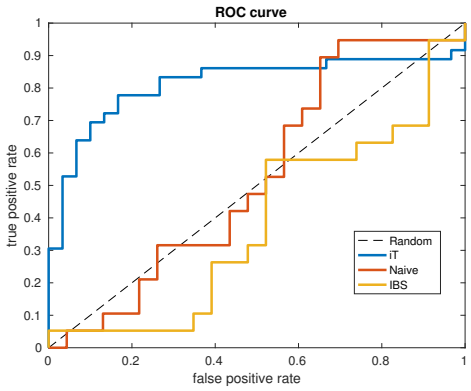


Fig. 9: Performance comparison considering the every-affordance case. The plot shows **iT** considerably outperforming the baseline methods with an accuracy of 80.30% and a precision of 84.85%, with a prediction threshold of 0.52.

D. Simulations

We demonstrate the applicability of our system in robotic scenarios implementing the system within the ROS [27] framework. Using the same synthetic scenes we construct a world for the robot in which we query affordance locations accordingly. Notice that for these simulations the input to our algorithm is no longer a full mesh or CAD model but a pointcloud that the robot captures with its sensor. Figure 10 illustrates examples of the simulation for *filling* a mug and *sitting*.

E. Real RGB-D Scenes

We conducted experiments on pointclouds captured with a Asus Xtion sensor using a publicly available dense mapping

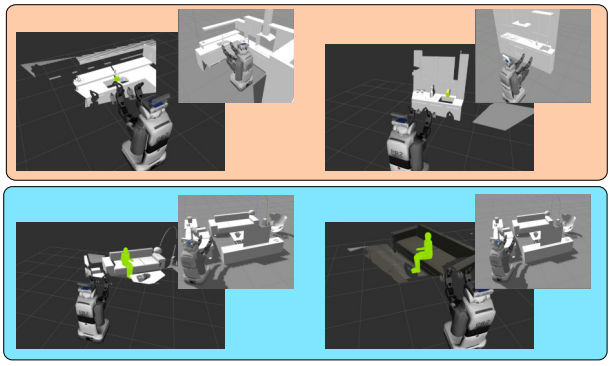


Fig. 10: The input to our algorithm in simulations is the pointcloud captured by the robot's sensor. In green is shown the query-object. Top row: best candidate location for *filling* a mug in a kitchen. Bottom row: best candidate location for *sitting* a human in a living room.

system [28]. Additionally, we included publicly available data containing high-quality and clean scans: the indoor scene scans from [29]; and 5 real-motorcycle scans from [30] in order to test our *riding* detection. This lead to a testing dataset comprised by 20 real scenes. Using the same pipeline explained before, we query object-affordance pairs for each of these scenes using the *training* example from the synthetic training data. The only pre-processing step carried out to these scenes is the ground plane calibration. Fig. 8 shows affordance heat-maps for these scenes and examples of the predicted locations.

V. DISCUSSION AND CONCLUSIONS

This paper presents and evaluates a new tensor field representation to express the *geometric* affordance of one object relative to another. By expanding the bisector surface representation to a richer tensor field, we are able to estimate

affordance locations on previously unseen scenes from a single example. The introduction of weighted tensor leads to affordance keypoints that allow faster decisions per query point and a compact and straight forward way to compute a descriptor. Our evaluation is carried out with both synthetic and real RGB-D scenes. The performance of our interaction tensor is significantly better in agreeing with crowdsourced opinions than the results of the baseline methods.

Our current approach relies on a known query-object whose 3D model is fully available, this limitation is due to the fact that incomplete objects in the *training* phase would cause artifacts in the bisector surface such as dips, spikes or object penetrations that could change the geometry of the **iT**. We do not think is inconceivable to have prior knowledge regarding query-object geometries; however, one possible avenue for future work is to investigate how to deal with partially perceived objects. One step towards these scenarios could be to replace the CAD models with RGB-D scans of real objects such as those available in [31], or furthermore exploring approaches similar to [32] in order to complete or approximate the unobserved geometry.

Overall, we see this work as an effort to motivate further advancing of approaches in Vision which, such as Active Perception [33], are more *ecological* in nature and consider the needs of the perceiving agent.

REFERENCES

[1] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, 1977.

[2] W. Warren, "Does this computational theory solve the right problem? Marr, Gibson, and the goal of vision," *Perception*, vol. 41, no. 9, pp. 1053–1060, 2012.

[3] H. Min, C. Yi, R. Luo, J. Zhu, and S. Bi, "Affordance Research in Developmental Robotics: A Survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 237–255, Dec 2016.

[4] A. Antunes, L. Jamone, G. Saponaro, A. Bernardino, and R. Ventura, "From human instructions to robot actions: Formulation of goals, affordances and probabilistic planning," in *Robotics and Automation (ICRA),IEEE International Conference on*, May 2016, pp. 5449–5454.

[5] G. Saponaro, P. Vicente, A. Dehban, L. Jamone, A. Bernardino, and J. Santos-Victor, "Learning at the Ends: From Hand to Tool Affordances in Humanoid Robots," in *Development and Learning and Epigenetic Robotics (ICDL-Epirob), Joint IEEE International Conferences on*, Sept 2018.

[6] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Self-supervised learning of grasp dependent tool affordances on the iCub Humanoid robot," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2015, pp. 3200–3206.

[7] G. Saponaro, G. Salvi, and A. Bernardino, "Robot anticipation of human intentions through continuous gesture recognition," in *Collaboration Technologies and Systems (CTS), International Conference on*, May 2013, pp. 218–225.

[8] A. Pandey and R. Alami, "Affordance graph: A framework to encode perspective taking and effort based affordances for day-to-day human-robot interaction," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Nov 2013, pp. 2180–2187.

[9] A. Pieropan, C. Ek, and H. Kjellstrom, "Functional object descriptors for human activity modeling," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2013, pp. 1282–1289.

[10] H. Koppula and A. Saxena, "Physically Grounded Spatio-temporal Object Affordances," in *Computer Vision ECCV 2014*. Springer International Publishing, 2014, vol. 8691, pp. 831–847.

[11] A. Srikantha and J. Gall, "Discovering object classes from activities," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 415–430.

[12] W. Chan, Y. Kakiuchi, K. Okada, and M. Inaba, "Determining proper grasp configurations for handovers through observation of object movement patterns and inter-object interactions during usage," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Sept 2014, pp. 1355–1360.

[13] Y. Jiang and A. Saxena, "Modeling High-Dimensional Humans for Activity Anticipation using Gaussian Process Latent CRFs." in *Robotics: Science and Systems*, 2014, pp. 1–8.

[14] C. Desai and D. Ramanan, "Predicting Functional Regions on Objects," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, June 2013, pp. 968–975.

[15] Y. Zhu, A. Fathi, and L. Fei-Fei, "Reasoning about Object Affordances in a Knowledge Base Representation," in *Computer Vision ECCV 2014*. Springer International Publishing, 2014, vol. 8690, pp. 408–424.

[16] Y. W. Chao, Z. Wang, R. Mihalcea, and J. Deng, "Mining semantic affordances of visual object categories," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4259–4267.

[17] A. Aldoma, F. Tombari, and M. Vincze, "Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2012, pp. 1732–1739.

[18] L. Hinkle and E. Olson, "Predicting object functionality using physical simulations," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Nov 2013, pp. 2784–2790.

[19] D. Kim and G. Sukhatme, "Semantic labeling of 3D point clouds with object affordance for robot manipulation," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2014, pp. 5578–5584.

[20] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance Detection of Tool Parts from Geometric Features," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2015.

[21] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Detecting object affordances with Convolutional Neural Networks," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Oct 2016, pp. 2765–2770.

[22] A. Roy and S. Todorovic, "A Multi-scale CNN for Affordance Segmentation in RGB Images," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 186–201.

[23] L. Piyathilaka and S. Kodagoda, "Affordance-map: Mapping human context in 3D scenes using cost-sensitive SVM and virtual human models," in *Robotics and Biomimetics (ROBIO) IEEE International Conference on*, Dec 2015, pp. 2035–2040.

[24] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert, "From 3D Scene Geometry to Human Workspace," in *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[25] X. Zhao, H. Wang, and T. Komura, "Indexing 3D Scenes Using the Interaction Bisector Surface," *ACM Trans. Graph.*, vol. 33, no. 3, pp. 22:1–22:14, June 2014.

[26] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Robotics and Automation (ICRA), IEEE International Conference on*, Shanghai, China, May 9-13 2011.

[27] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *Robotics and Automation Workshop (ICRA) International Conference on*, May 2009.

[28] S. Li and A. Calway, "RGBD relocalisation using pairwise geometry and concise key point sets," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2015, pp. 6374–6379.

[29] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 112:1–112:8, July 2013.

[30] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun, "A Large Dataset of Object Scans," *arXiv:1602.02481*, 2016.

[31] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and Model set: Towards common benchmarks for manipulation research," in *Advanced Robotics (ICAR) International Conference on*, July 2015, pp. 510–517.

[32] M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow, "Structured Prediction of Unobserved Voxels From a Single Depth Image," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, Feb 2018.